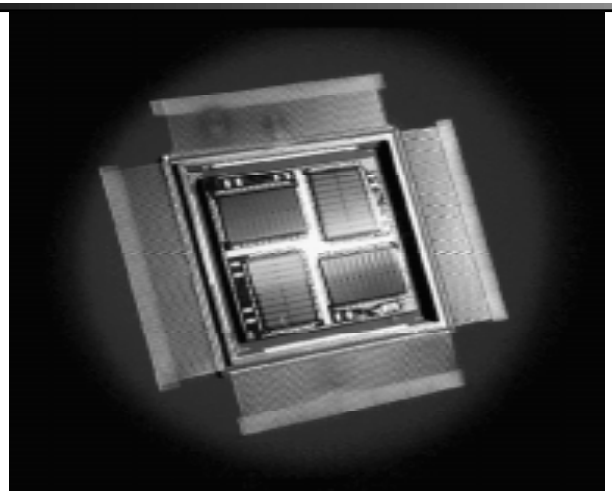
100 GOPS Reconfigurable Desktop System for Embedded Applications

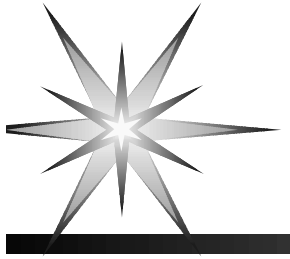


- Multi-million gates reconfigurable desktop machine based on FPMCM
- C++ design entry bit-serial synthesis system
- Visual programming for image processing under Khoros system

- Reconfigurable system hardware:
  - 1.6M-gate PC add-on board.
  - 6M-gate desktop.
- A standardizable programming model
- Solve defense computing's grand challenge problems (SAR, ATR).
  Ex. 2D FFT on 4Kx4K images in 1 sec

- FPMCM-III(400k gates) design: 06/98
- 1.6M-gate board design: 06/99
- Retarget C++ compiler system to Khoros: 06/98
- Image processing library & grand challenge applications: 06/98

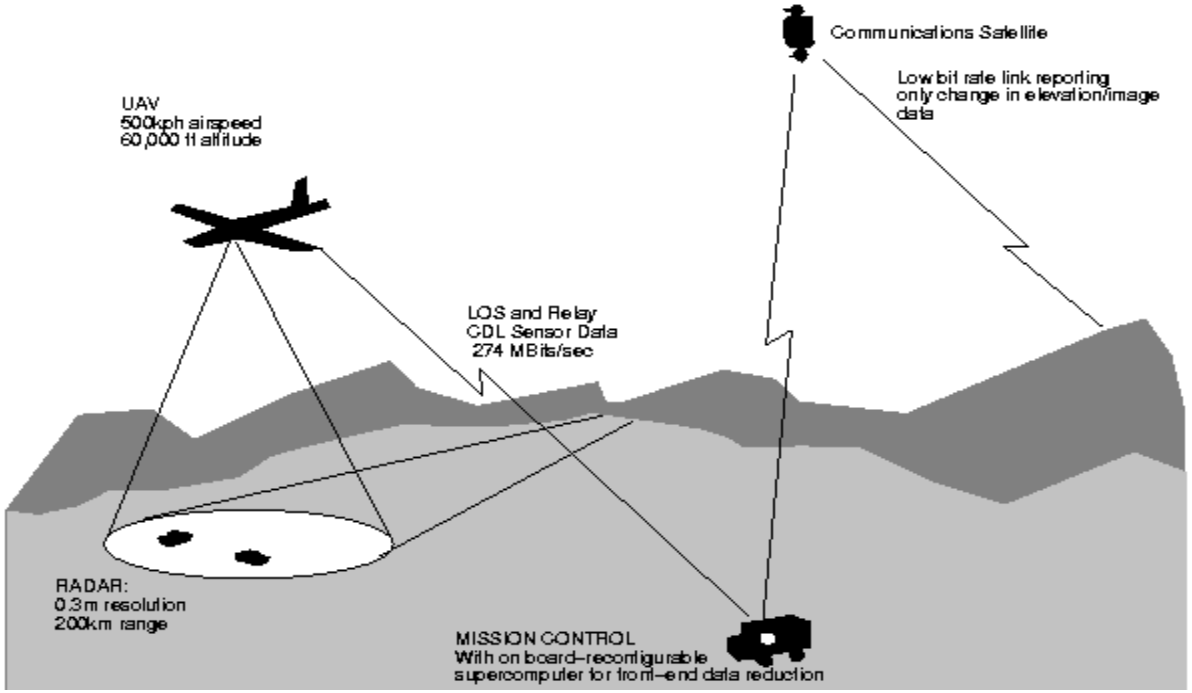
PI: Prof. Wayne Dai @UC Santa Cruz



Outline

- Motivation and Objectives
- Accomplishments: FPMCM-I, FPMCM-II, PCI Board, Reconfigurable System.
- Future Work: Hardware, Software, Applications
- Demo

Motivation



TIER-2 surveillance system(272Mbit/sec)

- Solve large scale image processing problems.

Case study: 2-D FFT on a 16 million samples image (4096x4096 32-bit) in 1 second: 64Mbyte/sec

Why 6M-Gate Desktop

- 2D FFT on 4096x4096 images (32-bit/pixel):
  - Mul = 16 $N^2 \log_2 N = 3 \times 2^{30}$

    Add = $4N^2 \log_2 N = 0.75 \times 2^{30}$ **in 1 second!**
  - Bases on bit-serial circuit, a 32-bit multiplier:
    - needs 130 LE (logic element: 4-LUT and flip-flop);
    - sampling rate is 2MHz.
  - Result: Only consider multiplier, we need

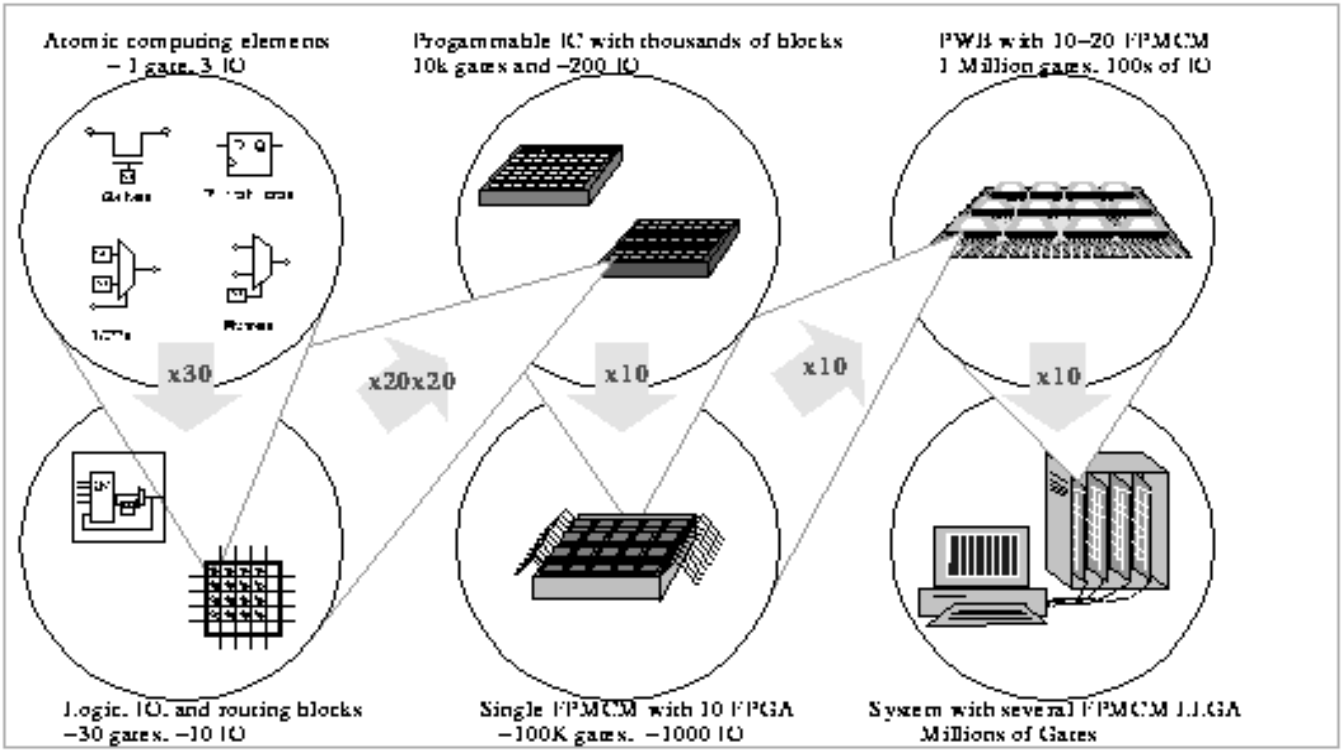
    # of LE = 209,379

    If we use Flex10K100 (5000LEs): 41 FPGAs
  - Consider adders and communications: 60 FPGAs.

Objectives

- **Problem**: FPGA-based systems today are bulky, slow and expensive.
- **Solution**: Deliver more compact, more-tightly coupled FPGA-systems using multi-chip modules and vertical interconnect.
- Objective: Demonstrate the revolutionary potential of MCM technology to create massive FPLDs.

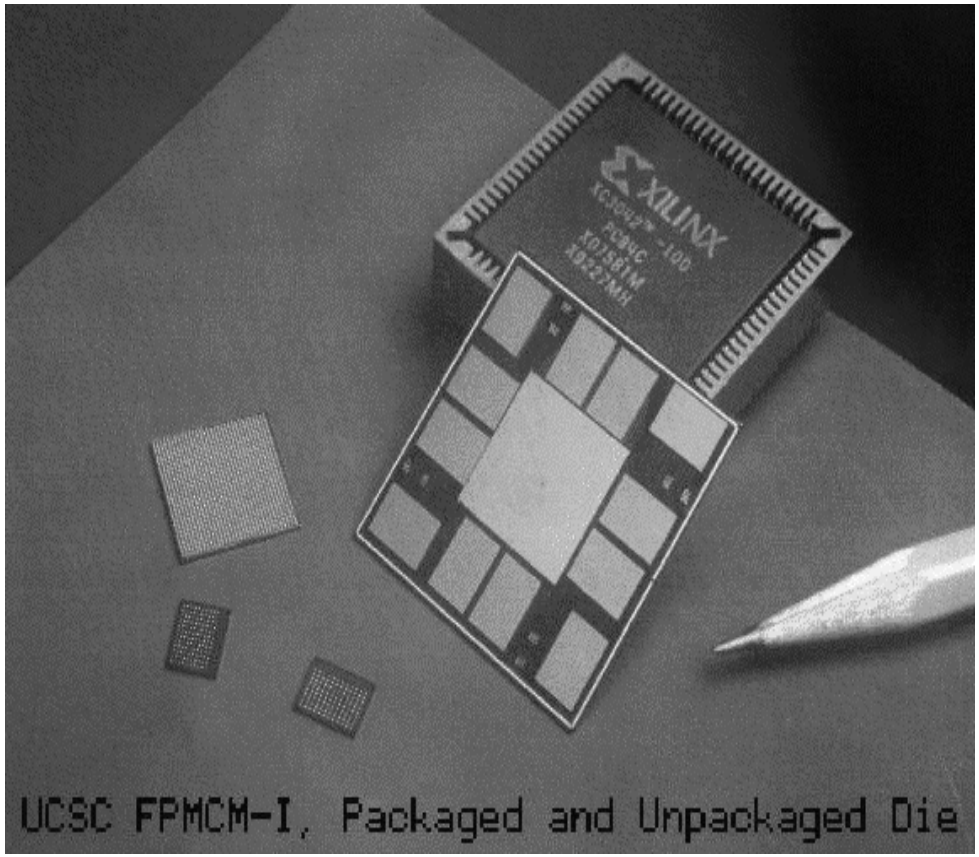
Objectives



- **Approach**: build next 2 generations of devices, ending in 6 million gates FPMCM box..

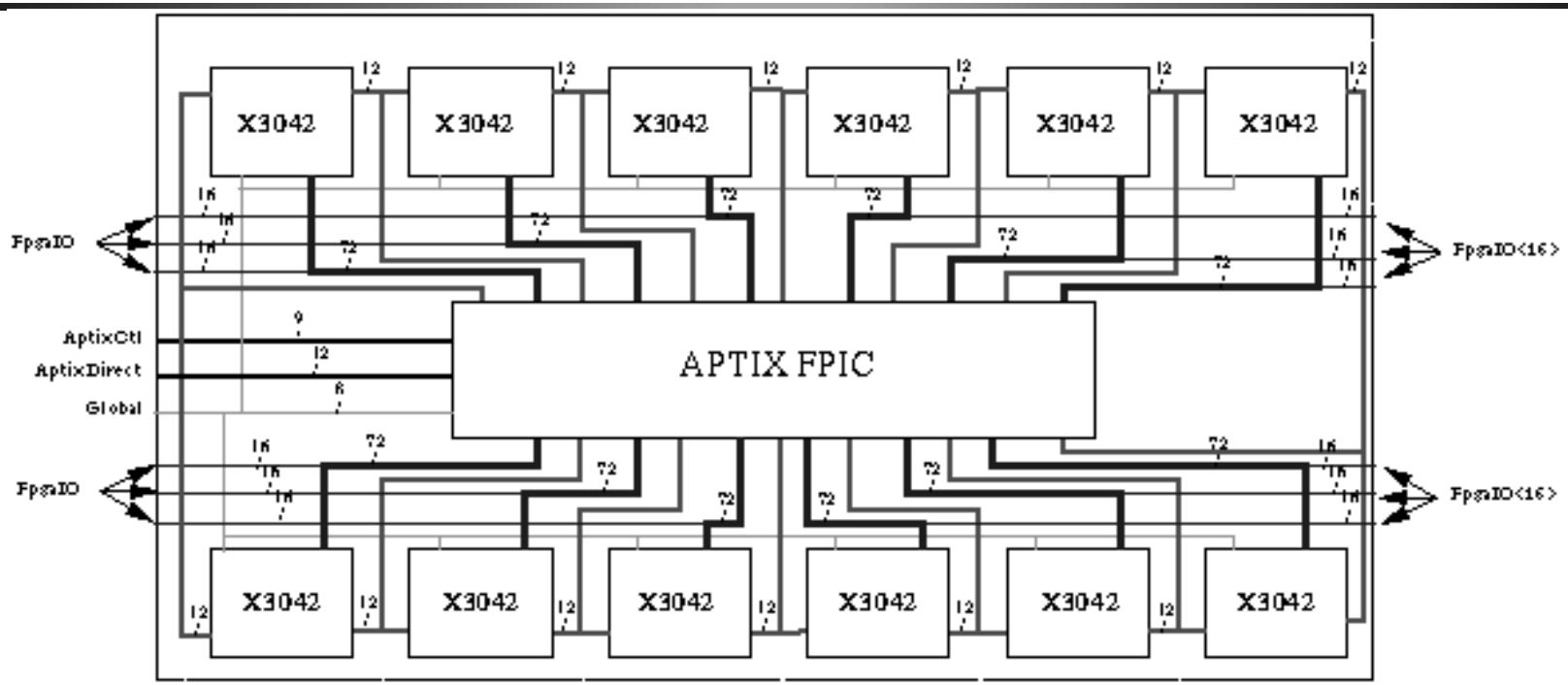
Accomplishment: FPMCM-I

- 44k gates FPMCM-I
- 12 Xilinx 3042 FPGAs connected by an Aptix FPIC with 1024 IOs.
- FPGAs were rerouted for low-cost flip-chip.
- 27MHz peak clock rate.
- 256 user IO.
- ~~Silicon efficiency: 70%.~~



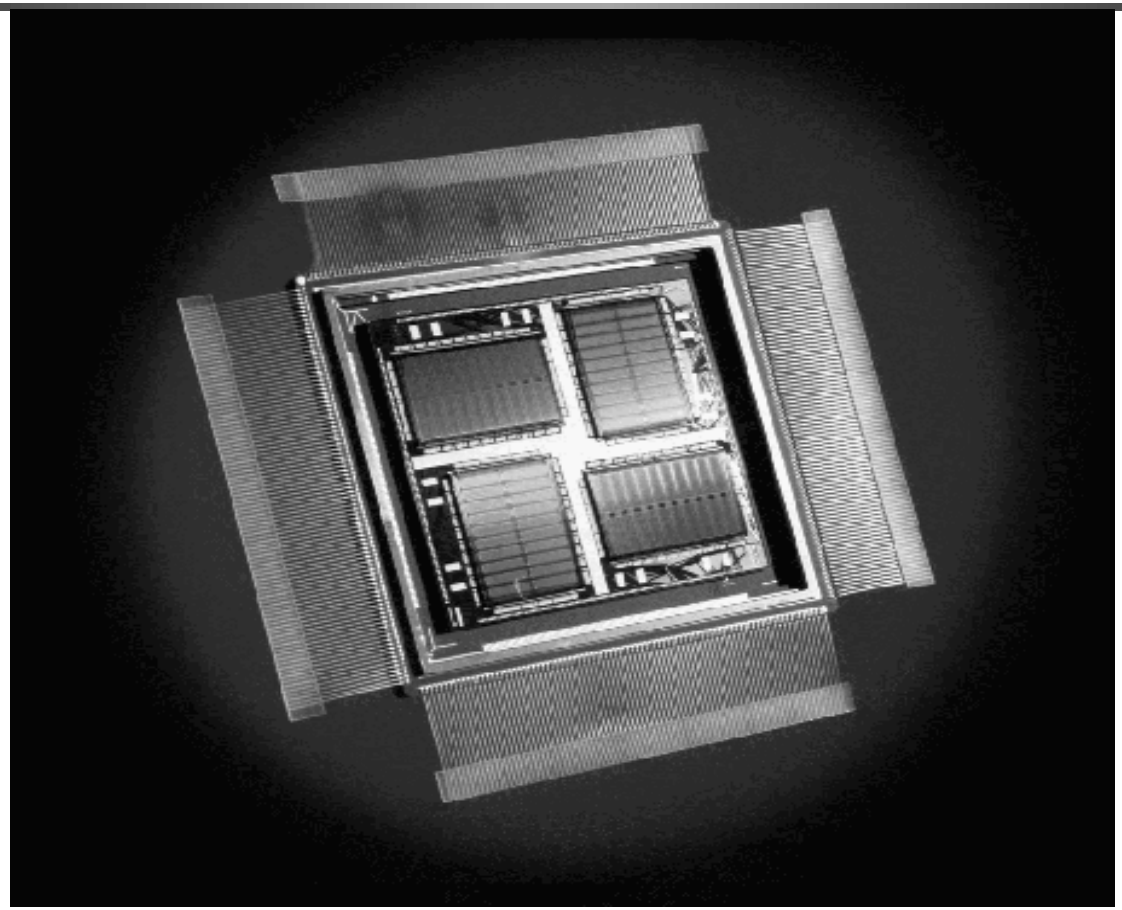
First silicon-on-silicon FPMCM

FPMCM-I Architecture



- Star architecture with some direct interconnect.
- Medium-sized FPGAs chosen to minimize cost per logic resource.

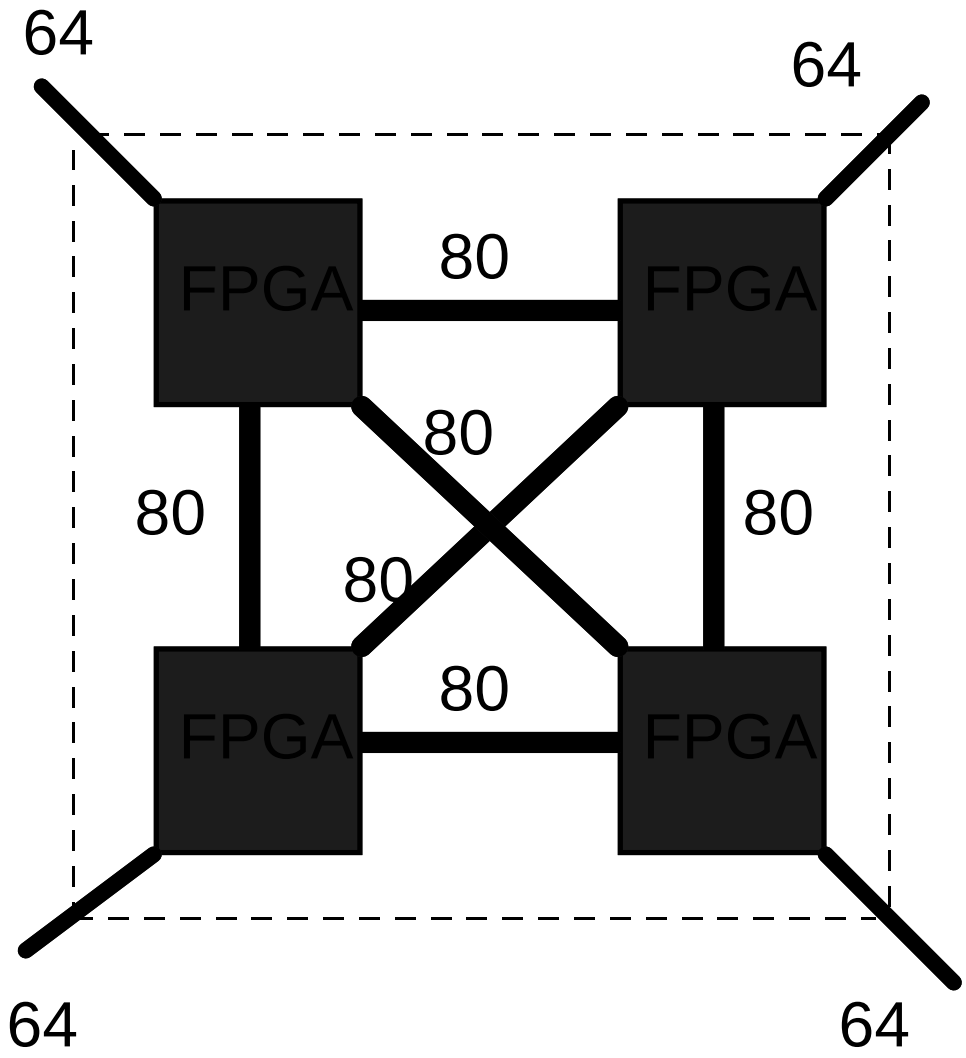
Accomplishment: FPMCM-II



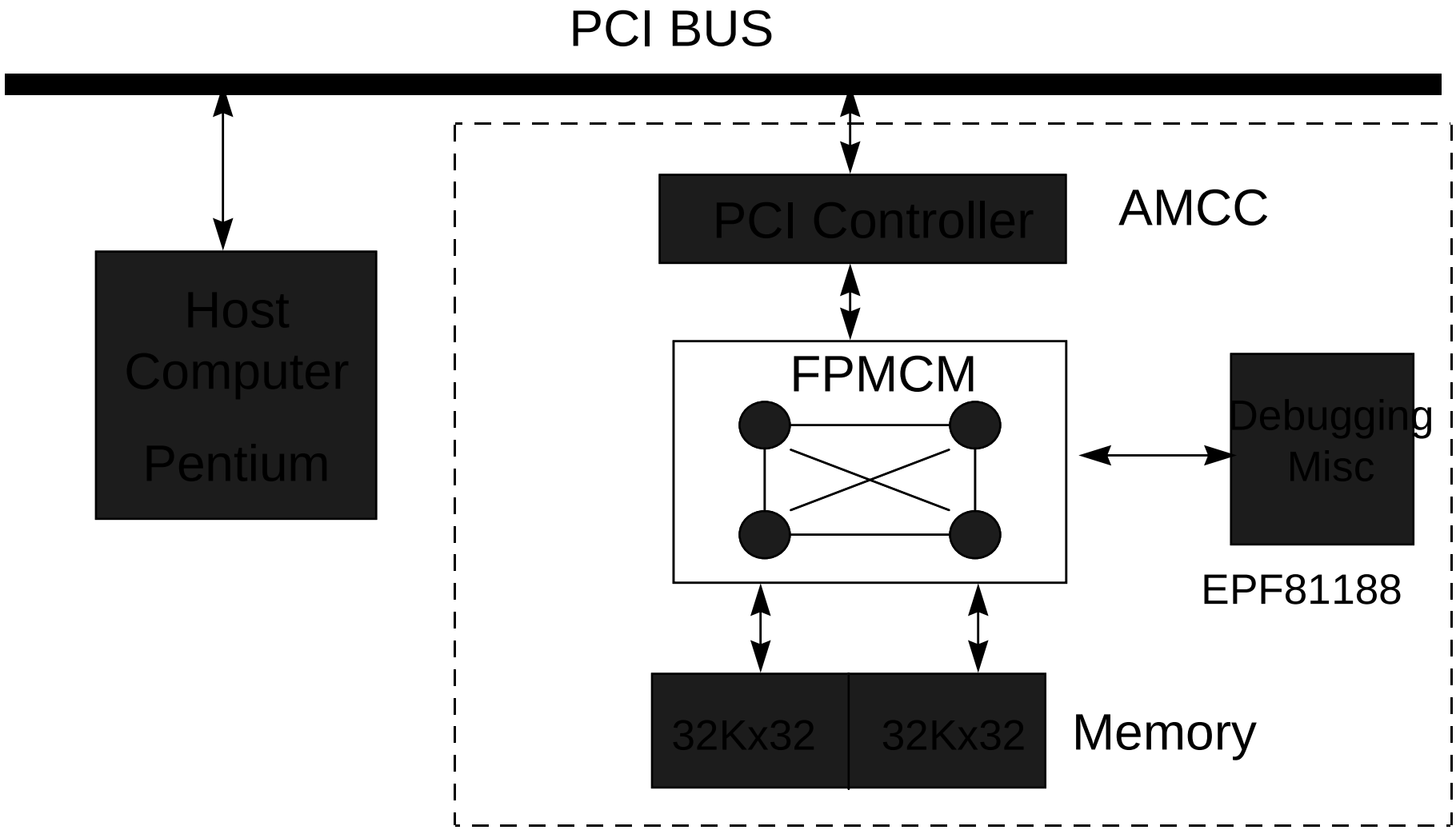
- 200k gates FPMCM-II: Composed of 4 Altera FLEX10K50.

FPMCM-II Architecture

- Clique Architecture
- Internal: 80 bits wide
- 64-bit to external
- Omit interconnection chip such as Aptix in FPMCM-I
- Drawback: fixed connection



Accomplishment: PCI Board

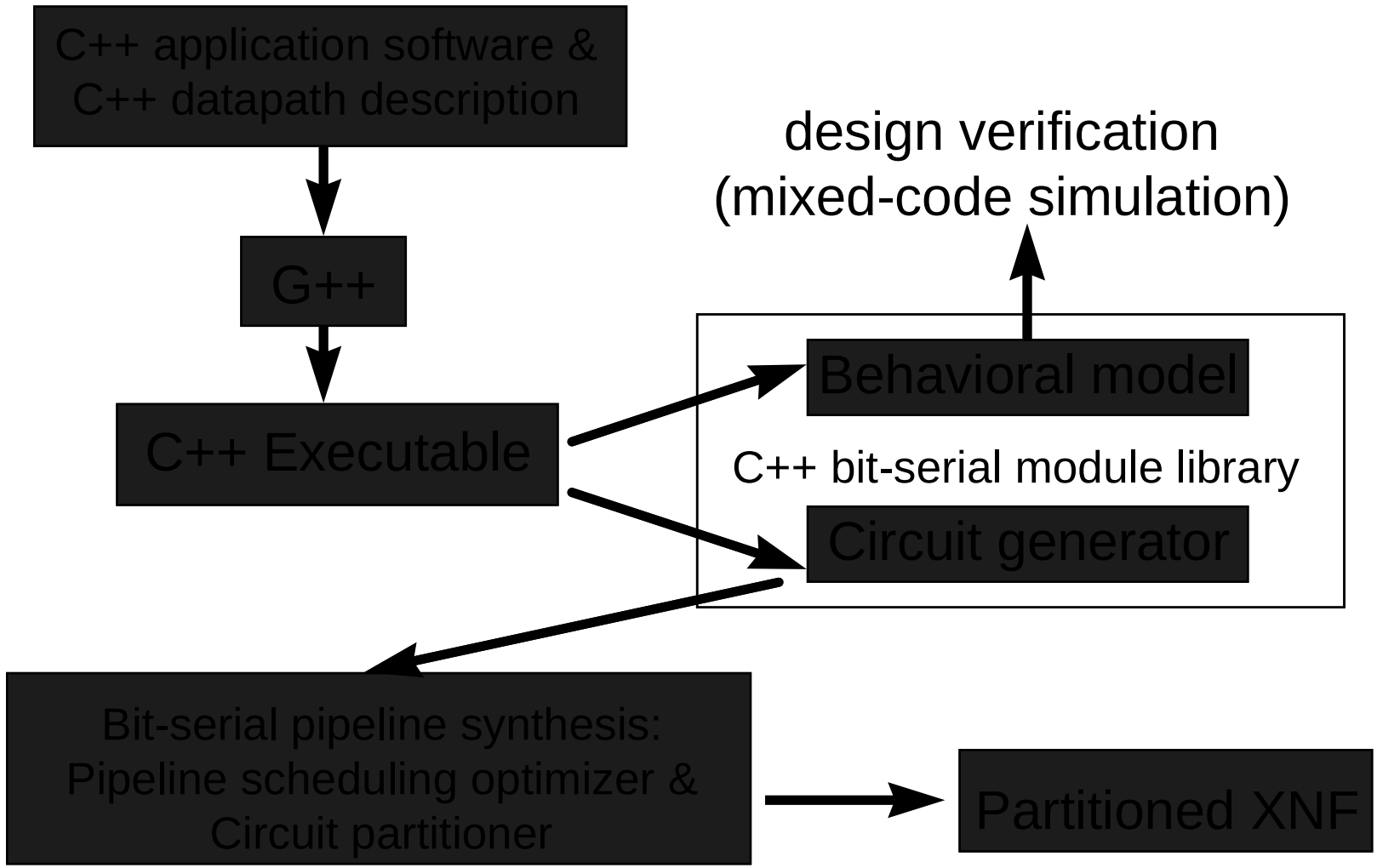


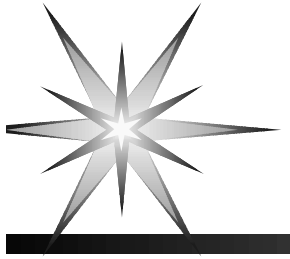
Board Architecture

Board Design

- The board can be plugged in the PCI bus of the popular PC (Pentium 166).
- The commercial PCI controller AMCC is used to reduce the risk during the early development.
- Two Synchronous memory chips are connected to the FPMCM.
- A standalone FPGA is used to implement various diagnostic and monitoring functions.

Accomplishment: C++ Bit Serial Synthesis System

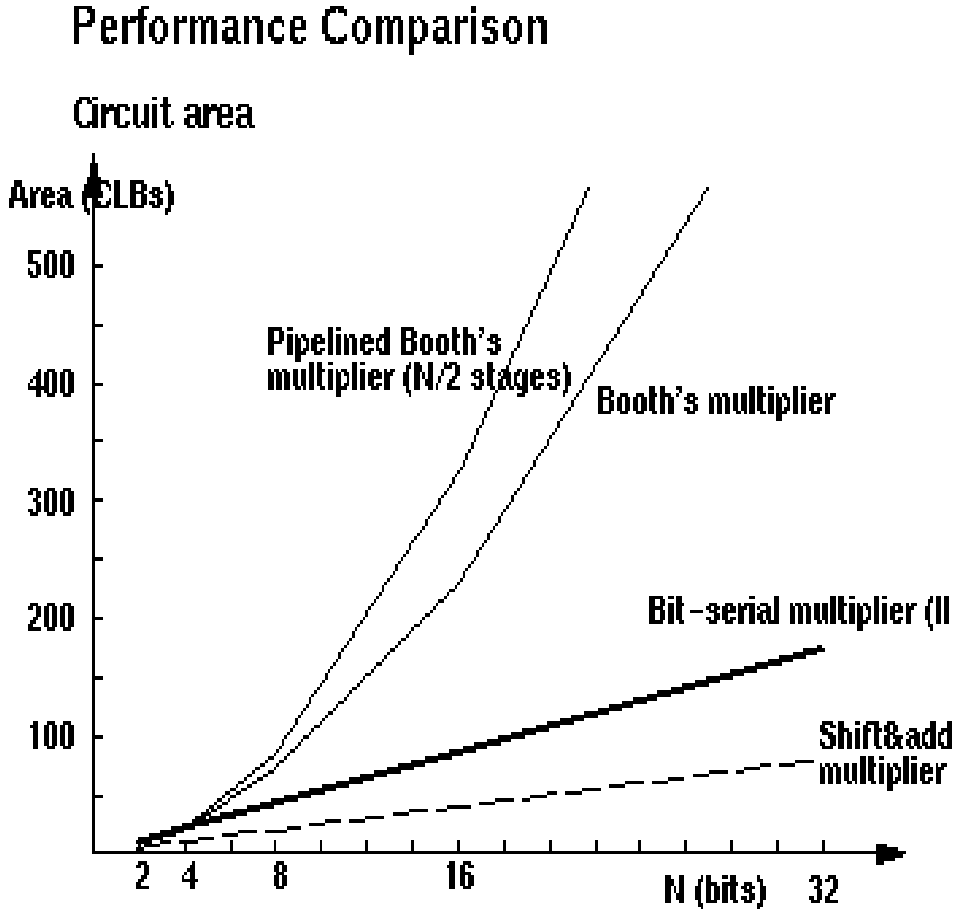


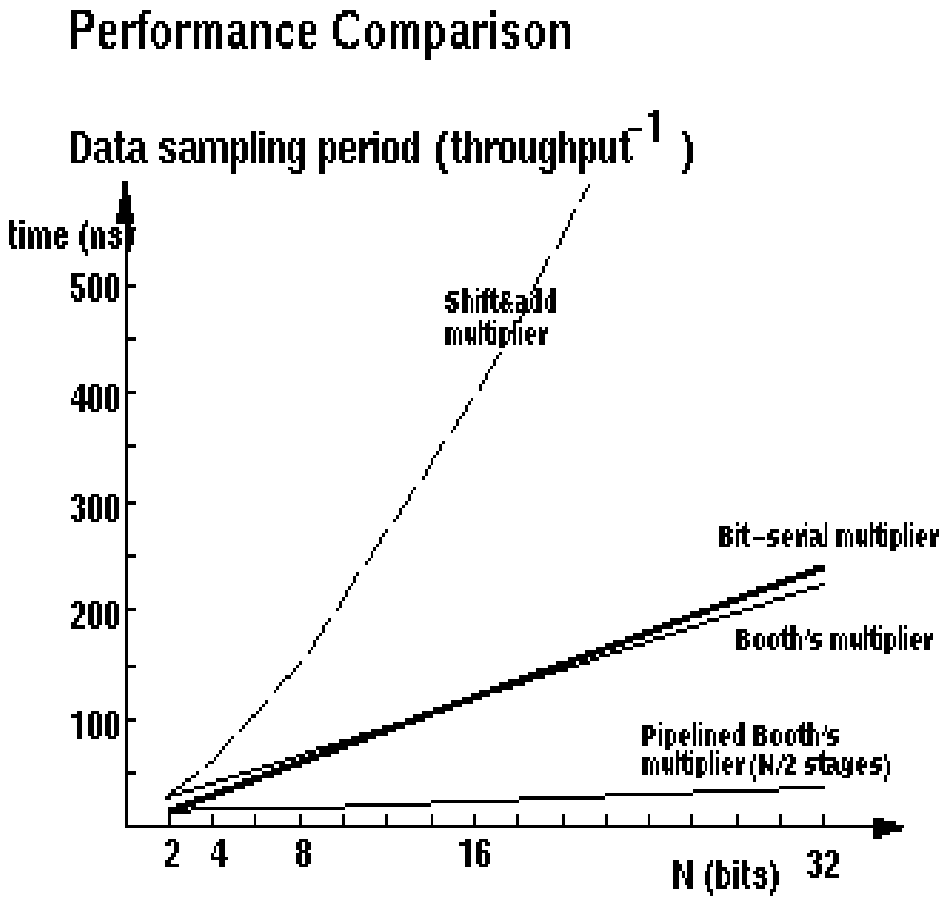


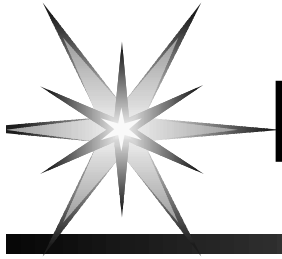
Why Bit Serial

- Most FPGAs are register rich. Bit serial pushes routing away from "x" and "y" dimensions into the "time" dimension. Place and route becomes easy.
- By not having to worry about routing delays, macros can be "soft" instead of "hard".
- Time-multiplexing makes the chip boundary less restrictive.
- FPGA clock periods are heavily routing limited. Bit serial logic naturally pipelines the communication for minimum delays.

Why Bit Serial







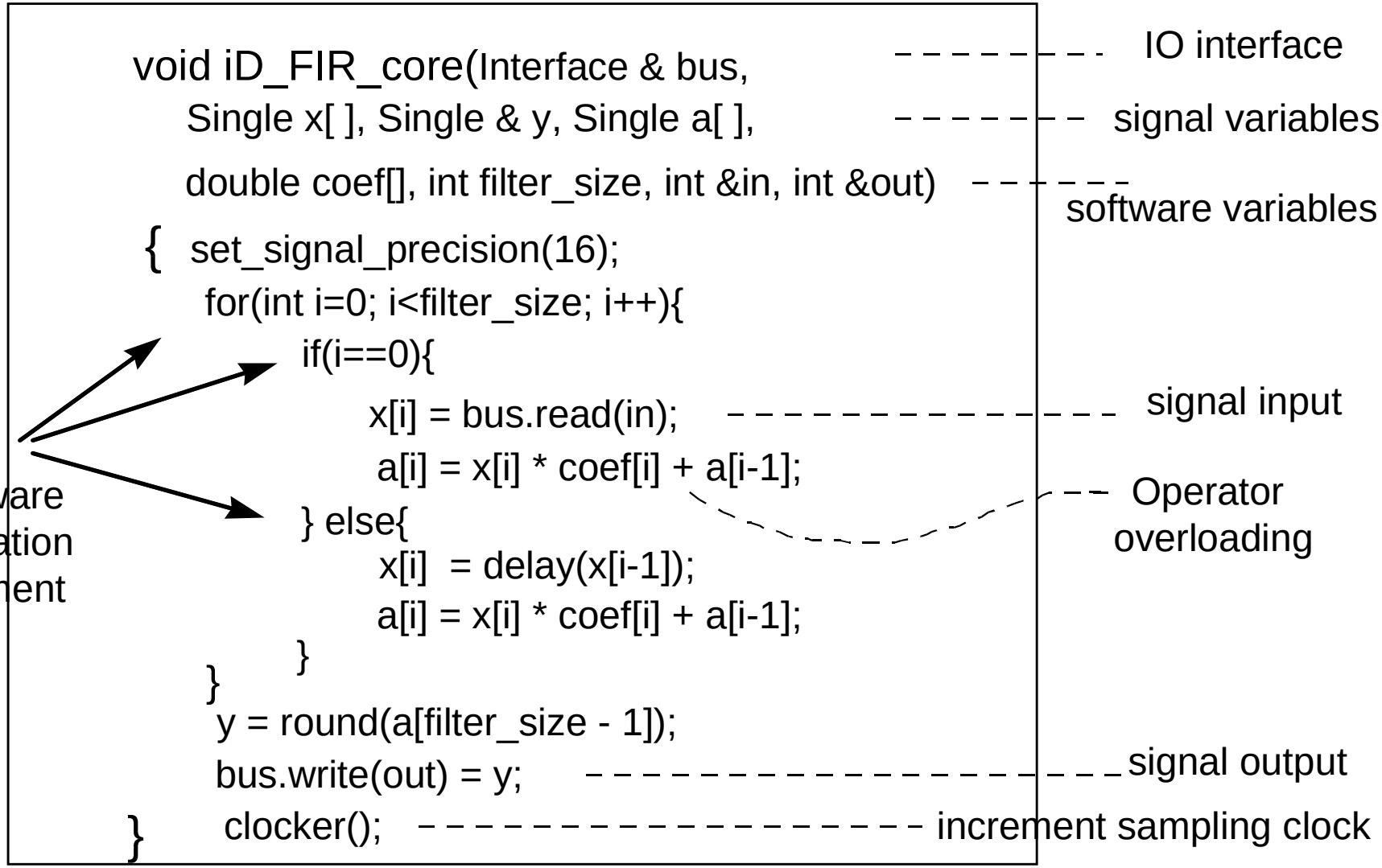
Bit-serial Circuits Design

- Highly efficient bit-serial circuits have been manually developed:
  - adder, substractor, multiplier, comparator, shifter, rounder, and saturater.

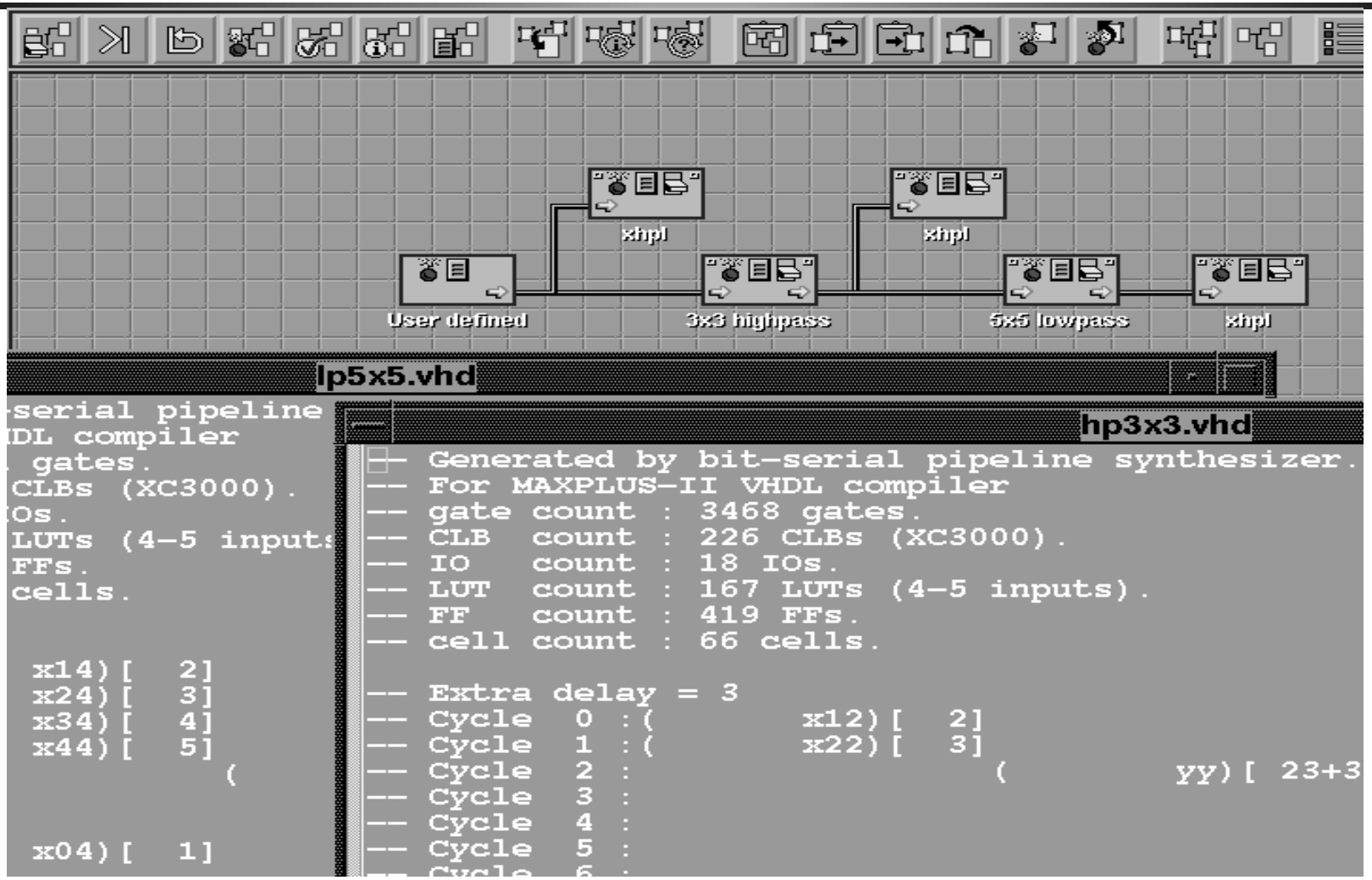
- Portable, open standard for user-generated application specific libraries.

C++ Design Capture

- C++ description example (1D FIR filter):

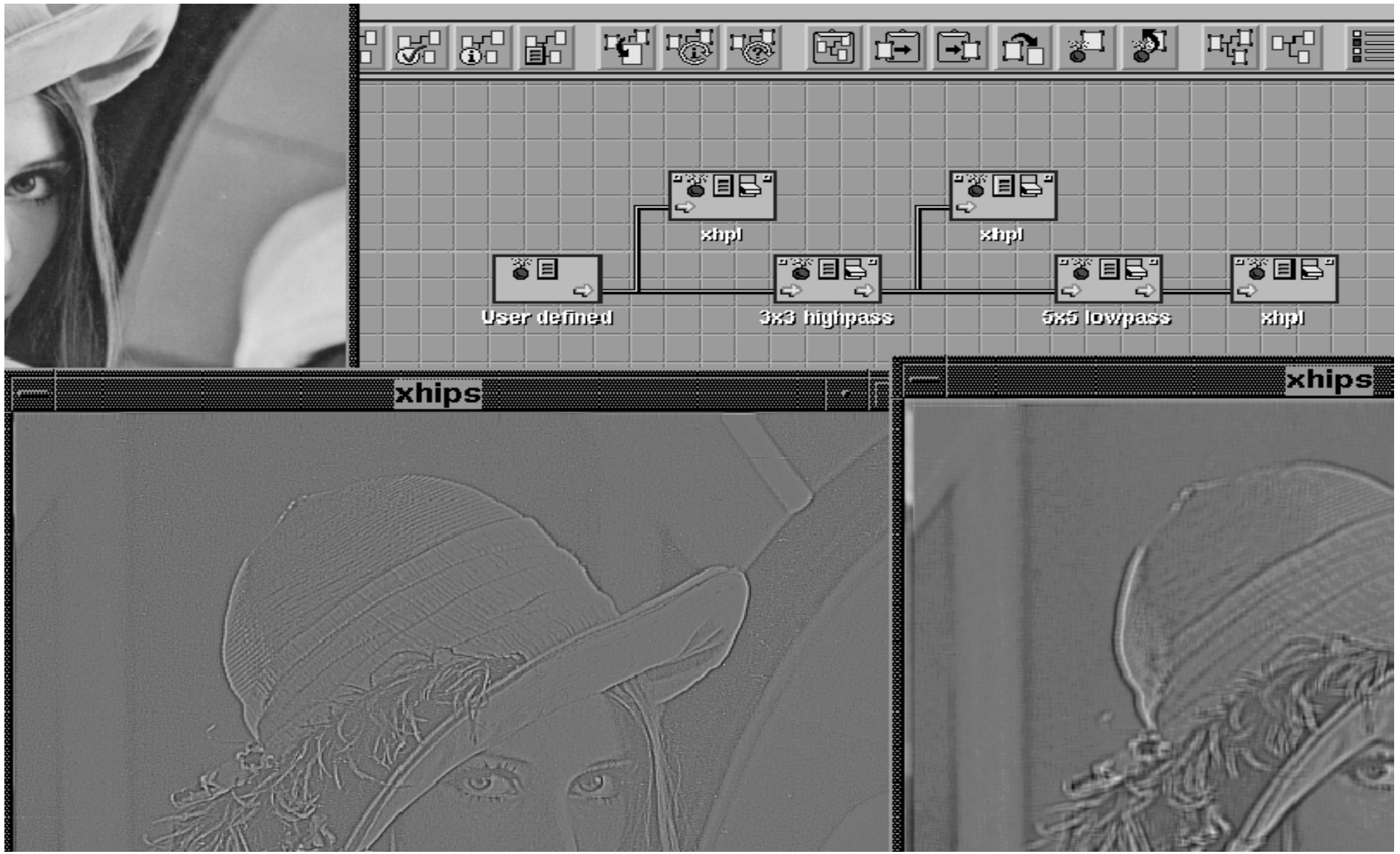


Visual Programming Based on Khoros

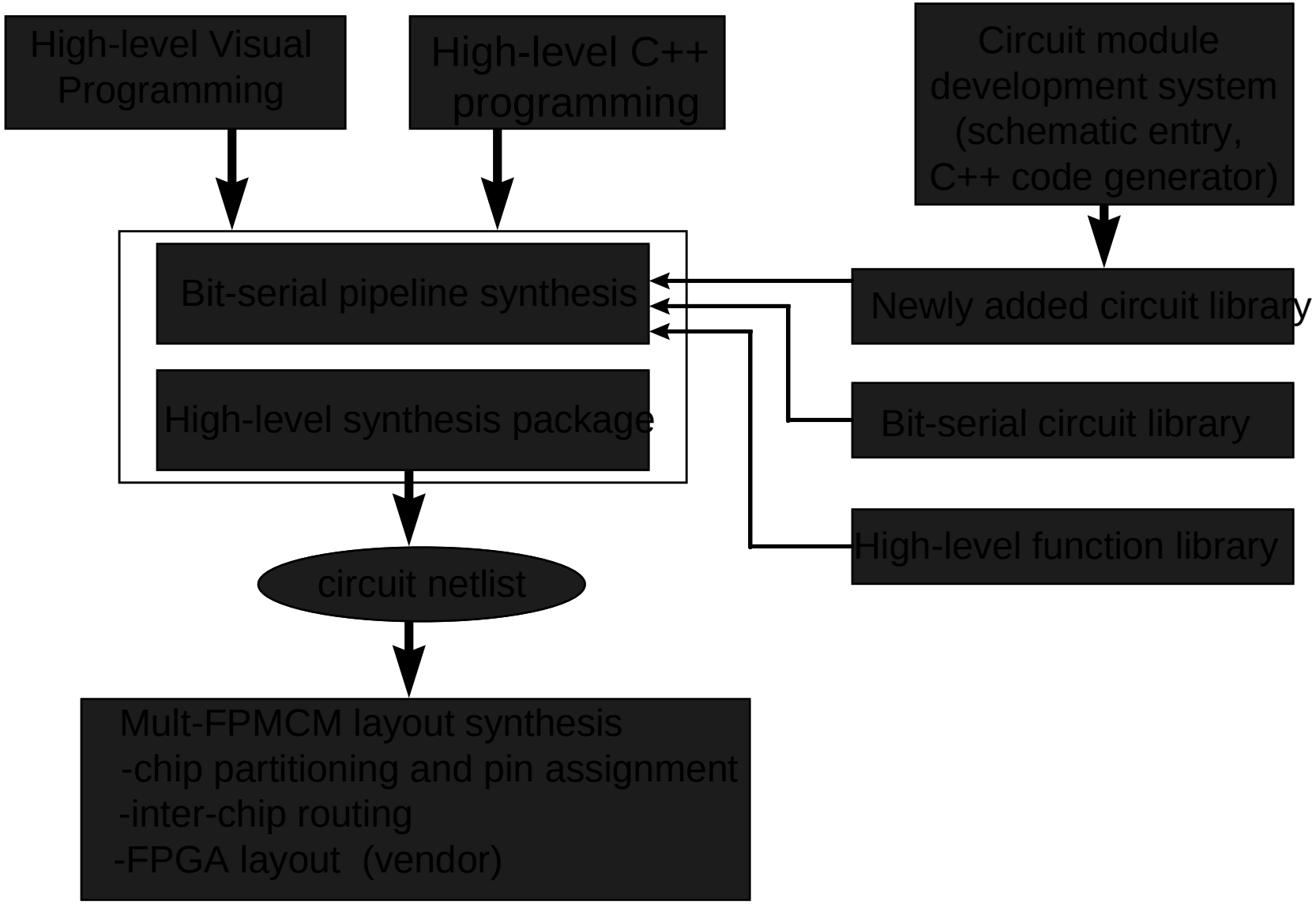


Input glyphs, then get VHDL description

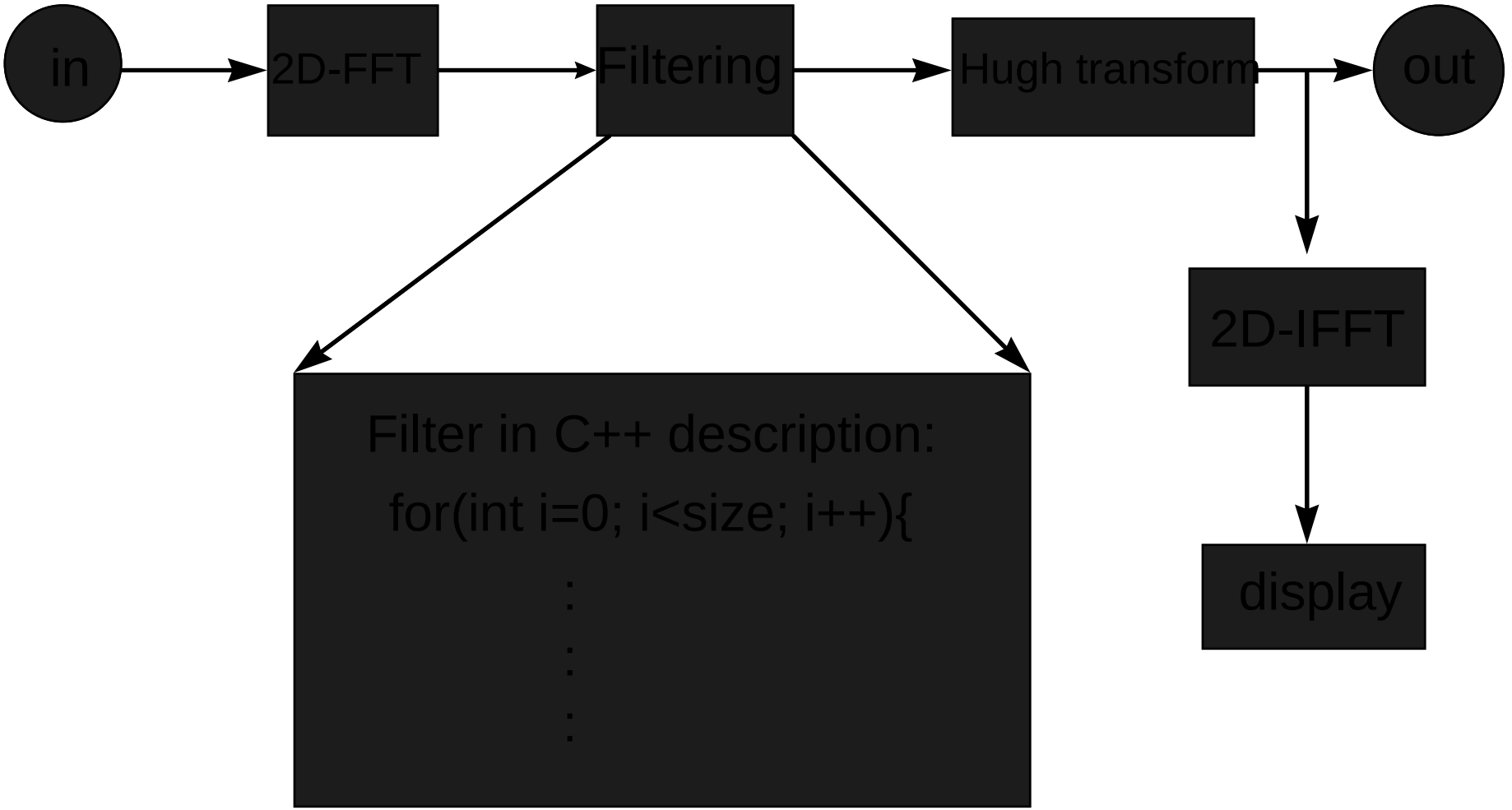
Visual Programming



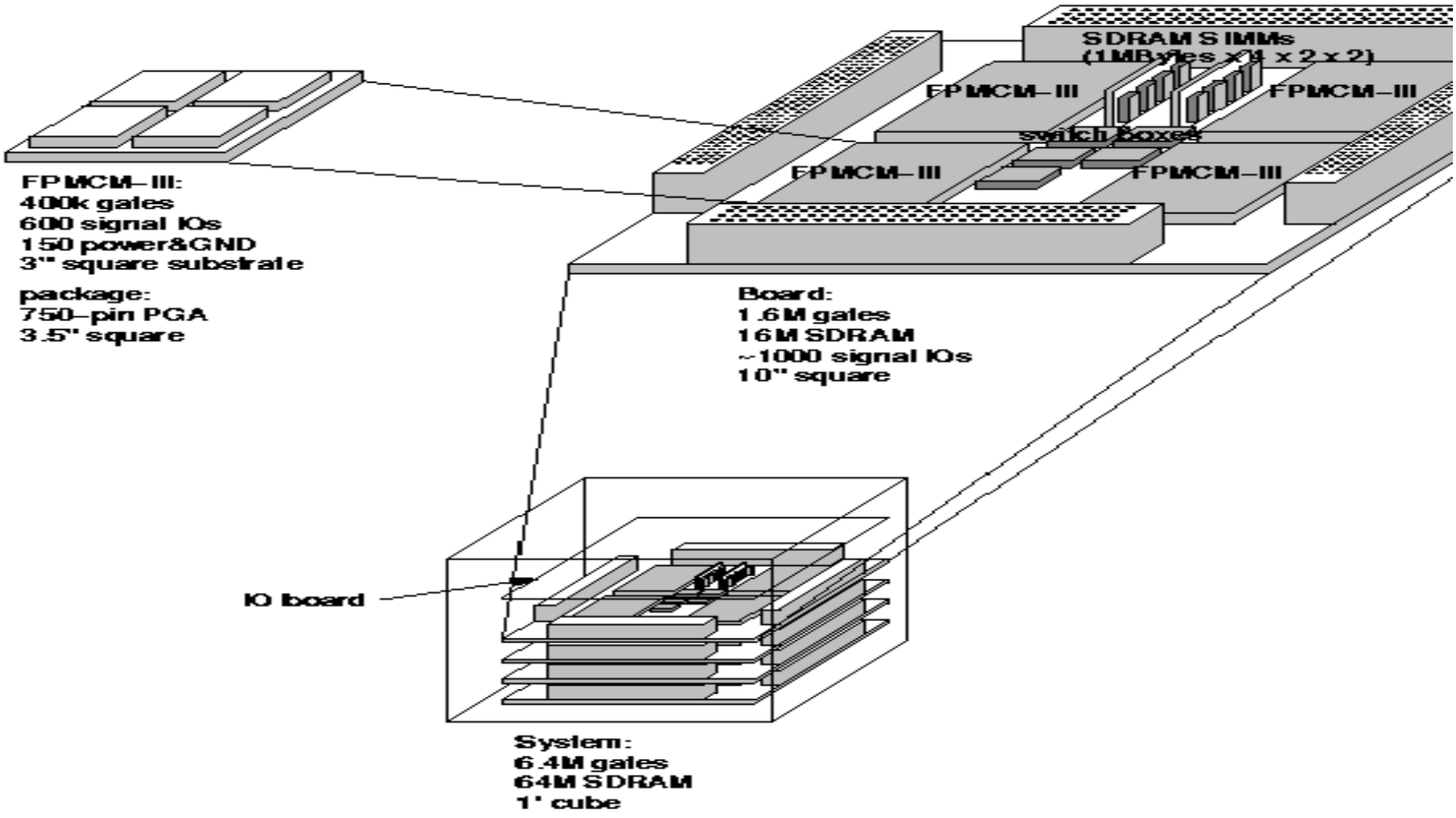
Future: Software System



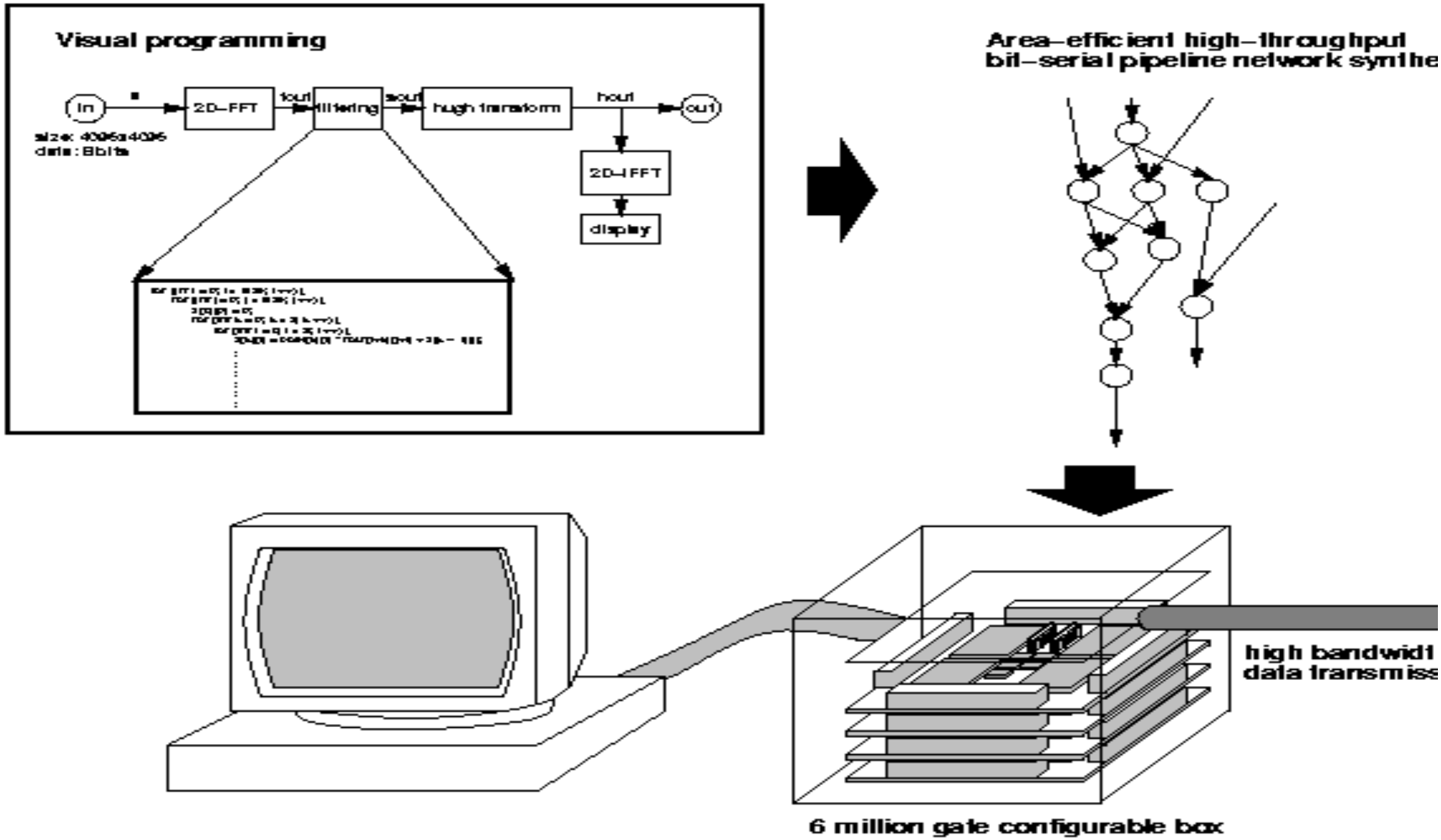
Future: Visual Programming

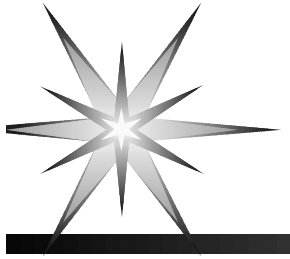


Future: Hardware Architecture



Future: Desktop System





Conclusions

- Reconfigurable Computing can approach custom hardware performance, and can be re-targeted to specific applications quickly and efficiently.
- High routability and low IO consumption of bit-serial circuits are ideal for adoptive computing system.
- Multi-million Gates Reconfigurable System, with appropriate programming technology can outperform multiprocessor based systems on a large class of problems.